

```

1
2 #####
3 ##### System hardware info
4 #####
5
6 ##Speces - CPU
7 cmd_CPU_0 = 'lscpu' #General CPU specs
8 cmd_CPU_1 = 'cat /proc/cpuinfo' #Each individual cpu core specs
9
10 ##Specs - RAM
11 cmd_RAM_0 = 'sudo lshw -short -C memory' #RAM list summary
12 cmd_RAM_1 = 'dmidecode --type memory' #Full RAM specs
13 cmd_RAM_2 = 'grep MemTotal /proc/meminfo' #Total memory
14 cmd_RAM_3 = 'awk "/MemTotal/ {print $2}" /proc/meminfo' #Total memory
15
16 ##Specs - DISK
17 cmd_DISK_0 = 'fdisk -l | grep Disk' #Total hard disk space
18 cmd_DISK_1 = 'lshw -short -C disk' #Disk specs summary
19 cmd_DISK_2 = 'lsblk'
20
21 ##Specs - OS
22 cmd_OS_0 = 'cat /etc/os-release' #Operating system specs
23
24 ##Specs - HOST
25 cmd_HOST_0 = 'hostnamectl' #host machine info
26
27 ##Specs - SOFTWARE
28 cmd_SOFT_0 = 'sudo yum list installed | wc -l' #count the software installed
29 cmd_SOFT_1 = 'sudo yum list installed' #list all software installed -->
30 https://www.cyberciti.biz/faq/check-list-installed-packages-in-centos-linux/
31
32 ##Usage - CPU
33 cmd_CPU_use_0 = 'top -b -n1' #using 'top' wont work -->
34 https://stackoverflow.com/questions/25101619/reading-output-of-top-command-using-paramiko
35 cmd_CPU_use_1 = 'ps -eo %cpu,comm --sort=-%mem | head -n 10'
36
37 ##Usage - RAM
38 cmd_RAM_use_0 = 'free -h' #Total free and used memory
39 cmd_RAM_use_1 = 'ps -eo %mem,comm --sort=-%mem | head -n 10' #TOP 10 MEMORY-CONSUMING
40 PROCESSES
41
42 cmd_CPU_RAM_use_0 = 'ps -eo %cpu,%mem,comm --sort=-%mem | head -n 10'
43
44 ##Usage - DISK -->
45 https://www.tecmint.com/find-top-large-directories-and-files-sizes-in-linux/
46 cmd_DISK_use_0 = 'df -H --output=source,size,used,avail'
47
48 cmd_DISK_use_1 = 'du -a /etc/ | sort -n -r | head -n 10' #TOP 10 DISK-CONSUMING PROCESSES
49
50 cmd_DISK_use_2 = 'df -h /' #total disk of the root directory
51
52 #cmd_DISK_use_3 = 'df -h --output='size','pcent' /'
53 #https://linuxiac.com/check-disk-space-linux/
54
55 cmd_DISK_use_4 = 'df -h -t ext4'
56 #https://www.howtogeek.com/409611/how-to-view-free-disk-space-and-disk-usage-from-the-linux-terminal/
57
58 ##### SYSTEM & URL #####
59
60 ##System - PING
61 cmd_Sys_0 = 'ping -c 10 ' #must have a space after ping. server_ip is a variable
62 containing ip
63
64 ##System - Uptime
65 cmd_Sys_1 = 'uptime' #How long the machine has been up & running
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99

```

```

60
61 #####
62 ##### Linux packages
63 #####
64
65 # list last installed packages
66 cmd_installed_pkg = 'rpm -qa --last'
67
68 update_packages="yum -y update"
69
70
71 #####
72 ##### 7 zip
73 #####
74
75 ### 7zip for file zip -- https://elearning.wslidp.com/pcmagazine/extract-7zip-centos-7/
76 cmd_install_p7zip = 'yes | sudo yum install p7zip'
77
78 #https://unix.stackexchange.com/questions/470266/how-to-locate-a-package-installed-by-yum/470458
79 cmd_p7zip_install_path = 'rpm -q --filesbypkg p7zip'
80 cmd_search_7zip_version = 'yum search 7zip'
81 cmd_info_7zip_version = 'yum info 7zip'
82
83 cmd_remove_p7zip = 'yes | yum remove p7zip'
84 cmd_update_after_remove_7zip = 'yum update'
85
86 cmd_create_p7zip_TFM = ''
87
88
89 #####
90 ##### Epeel release
91 #####
92
93 cmd_install_epel_release_for_7zip = 'yes | sudo yum install epel-release' # yes | will
94 ansewr YES to the prompt. yes no | ansewr NO to the prompt.
95 cmd_import_epel_gpg_key = 'rpm --import
96 http://download.fedoraproject.org/pub/epel/RPM-GPG-KEY-EPEL-6'
97
98 cmd_search_epel_version = 'yum search epel-release'
99 cmd_epel_install_path = 'rpm -q --filesbypkg epel-release'
100
101 #Check installed package versions --
102 https://www.tecmint.com/install-epel-repository-on-centos/
103 cmd_info_epel_version = 'yum info epel-release'
104 cmd_remove_epel_release = 'yes | yum remove epel-release' # yes | will ansewr YES to the
105 prompt. yes no | ansewr NO to the prompt.
106 cmd_update_after_remove_epel_release = 'yes | yum update' # yes | will ansewr YES to the
107 prompt. yes no | ansewr NO to the prompt.
108
109
110 #####
111 ##### Iptables
112 #####
113
114 iptables_install="$SUDO $INSTALLER -y install iptables iptables-persistent"
115 ufw_install_for_centos_1="$SUDO $INSTALLER -y install epel-release"
116 ufw_install="$SUDO $INSTALLER -y install ufw"
117 iptables_version="iptables -V"
118 iptables_ufw_iptables_enable="$SUDO ufw --force enable"
119 iptables_ufw_iptables_disable="$SUDO ufw disable"
120 iptables_ufw_iptables_status="$SUDO ufw status"
121
122
123 #####
124 ##### Lsyncd

```

```

121 #####
122
123 cmd_install_rsync="yum install -y rsync"
124 cmd_lsyncd_install="yum -y install lsyncd"
125 cmd_install_sshclients="yum -y install openssh-clients"
126 cmd_rm_id_rsa='rm -f /root/.ssh/id_rsa'
127 cmd_ssh_keygen_create_keys='ssh-keygen -b 2048 -t rsa -f /root/.ssh/id_rsa -q -N ""'
128 cmd_start_lsyncd="systemctl start lsyncd"
129 cmd_stop_lsyncd="systemctl stop lsyncd"
130 cmd_enable_lsyncd="systemctl enable lsyncd"
131
132
133 #####
134 ##### Rsync
135 #####
136
137 rsync_install="$SUDO $INSTALLER -y install rsync"
138 rsync_version="rsync --version"
139 rsync_start="$SUDO service rsync start"
140 rsync_stop="$SUDO service rsync stop"
141 rsync_status="$SUDO service rsync status"
142 rsync_restart="$SUDO service rsync restart"
143 rsync_mkdir_www="$SUDO mkdir -p /tmp/var/www"
144 rsync_local_backup="$SUDO rsync -avn /var/www /tmp/var/www"
145 rsync_remote_backup= ''
146 #rsync_grep_www="$SUDO grep -nri "/tmp/var/www" /etc/crontab"
147 rsync_mode_change="$SUDO chmod 0666 /etc/crontab"
148 rsync_mode_revert="$SUDO chmod 0644 /etc/crontab"
149 rsync_copy_rsyncd_conf="$SUDO cp /usr/share/doc/rsync/examples/rsyncd.conf /etc"
150
151
152 #####
153 ##### Docker
154 #####
155
156 cmd_check_install_or_update_docker = ''' if [ -x "$(command -v docker)" ]; then
157     echo "Update docker"
158     # command
159 else
160     echo "Install docker"
161     # command
162 fi '''
163
164 docker_help ='docker --help'
165 docker_install = ''
166 docker_container_list = 'docker ps'
167 docker_version = 'docker version'
168 docker_status = 'service docker status'
169 docker_start =''
170 docker_stop = ''
171 docker_restart =''
172
173 cmd_Docker_install_path = 'rpm -q --filesbypkg docker' #this may not work in some
situtations
174
175 #remove docker to install a clean copy of docker
176 cmd_remove_docker = '''yum remove docker \
177     docker-client \
178     docker-client-latest \
179     docker-common \
180     docker-latest \
181     docker-latest-logrotate \
182     docker-logrotate \
183     docker-engine'''
184
185
186

```

```

187 cmd_install_docker = ''
188
189 # Multi-line bash script
190 cmd_Docker_running_or_not = '''if curl -s --unix-socket /var/run/docker.sock http/_ping
2>&1 >/dev/null
191 then
192     echo "Running"
193 else
194     echo "NOT running"
195 fi '''
196
197 cmd_Docker_status = 'systemctl status docker' #To check if Docker engine is running
198 cmd_start_docker_engine = 'systemctl start docker'
199 cmd_docker_version = 'docker version'
200
201 #Check if docker is installed, if so update it
202 cmd_check_install_or_update_docker = ''' if [ -x "$(command -v docker)" ]; then
203     echo "Update docker"
204     # command
205 else
206     echo "Install docker"
207     # command
208 fi '''
209
210 cmd_install_docker_engine = ''
211
212
213 # ##### Docker compose #####
214 cmd_docker_compose_version = 'docker-compose version'
215
216 cmd_install_docker_compose = ''
217
218 cmd_stop_docker_compose = 'docker-compose stop'
219 cmd_start_docker_compose = 'docker-compose up -d'
220
221
222
223 #####
224 ##### Php
225 #####
226
227
228
229
230
231
232
233
234
235 #####
236 ##### Joomla
237 #####
238
239 #https://www.digitalocean.com/community/tutorials/how-to-install-joomla-on-a-virtual-serv
er-running-centos-6
240
241 cmd_install_wget_unzip="yum install -y wget unzip"
242 cmd_download_joomla="wget -P /tmp/
https://downloads.joomla.org/cms/joomla3/3-9-25/Joomla_3-9-25-Stable-Full_Package.zip"
243 cmd_unzip_joomla="unzip -o /tmp/Joomla_3-9-25-Stable-Full_Package.zip -d /var/www/html/"
244 cmd_chown_html_dir="chown -R apache:apache /var/www/html/*"
245 cmd_chmod_html_dir="chmod -R 775 /var/www/html/*"
246 cmd_unzip_backup="unzip -o /tmp/joomla.zip -d /var/www/html/"
247 joomla_owner_permission_joomla="$SUDO chown -R www-data:www-data /var/www/html/joomla"
248 joomla_mode_changes_joomla="$SUDO chmod -R 755 /var/www/html/joomla"
249 #joomla_copy_file="$SUDO cp /tmp/$OS_TYPE/joomla.conf /etc/apache2/sites-available"
250 #joomla_enable_virtualhost_1="$SUDO a2ensite joomla.conf"

```

```

251 #joomla_enable_virtualhost_2="$SUDO a2enmod rewrite"
252
253
254 cmd_joomla_file_list = 'find / -name "'joomla'" '
255 cmd_admintools_file_list = 'find / -name "'admintools'" '
256 cmd_rsfirewall_file_list = 'find / -name "'rsfirewall'" '
257
258
259 #####
260 ##### Apache - Httpd
261 #####
262
263 cmd_httpd_install="yum install -y httpd"
264 cmd_httpd_stop="systemctl stop httpd"
265 cmd_httpd_enable="systemctl enable --now httpd"
266 cmd_http_firewall="firewall-cmd --permanent --zone=public --add-service=http"
267 cmd_https_firewall="firewall-cmd --permanent --zone=public --add-service=https"
268 cmd_php_chown="chown -R root:apache /var/lib/php/*"
269 cmd_restart_php="systemctl restart php-fpm.service"
270 cmd_install_epel="yum install -y epel-release yum-utils"
271 cmd_install_remi="yum -y install https://rpms.remirepo.net/enterprise/remi-release-7.rpm"
272 cmd_remi_disable="yum-config-manager --disable remi-php54"
273 cmd_remi_enable="yum-config-manager --enable remi-php74"
274 cmd_install_php_tools="yum -y update && yum -y install php-fpm php-cli php-gd
php-opcache php-mysqlnd php-json php-mcrypt php-xml php-curl mod_php"
275
276
277 #####
278 ##### Maria DB - Galera cluster
279 #####
280
281 cmd_mariadb_install="yum install -y MariaDB-server MariaDB-client galera"
282 cmd_mariadb_start="systemctl start mariadb"
283 cmd_mariadb_stop="systemctl stop mariadb"
284 cmd_mariadb_restart="systemctl restart mariadb"
285 cmd_mariadb_enable="systemctl enable mariadb"
286 cmd_galera_start="galera_new_cluster"
287 cmd_firewalld_add_3306__tcp_port="firewall-cmd --permanent --zone=public
--add-port=3306/tcp"
288 cmd_firewalld_add_4567__tcp_port="firewall-cmd --permanent --zone=public
--add-port=4567/tcp"
289 cmd_firewalld_add_4568__tcp_port="firewall-cmd --permanent --zone=public
--add-port=4568/tcp"
290 cmd_firewalld_add_4444__tcp_port="firewall-cmd --permanent --zone=public
--add-port=4444/tcp"
291 cmd_firewalld_add_4567__udp_port="firewall-cmd --permanent --zone=public
--add-port=4567/udp"
292 cmd_update_galera_node_status="sed -i 's/safe_to_bootstrap: 0/safe_to_bootstrap: 1/'
/var/lib/mysql/grastate.dat"
293
294
295 cmd_firewalld_add_3306__tcp_port = (
296     "firewall-cmd --permanent --zone=public --add-port=3306/tcp"
297 )
298 cmd_firewalld_add_4567__tcp_port = (
299     "firewall-cmd --permanent --zone=public --add-port=4567/tcp"
300 )
301 cmd_firewalld_add_4568__tcp_port = (
302     "firewall-cmd --permanent --zone=public --add-port=4568/tcp"
303 )
304 cmd_firewalld_add_4444__tcp_port = (
305     "firewall-cmd --permanent --zone=public --add-port=4444/tcp"
306 )
307 cmd_firewalld_add_4567__udp_port = (
308     "firewall-cmd --permanent --zone=public --add-port=4567/udp"
309 )
310 cmd_update_galera_node_status = (

```

```

311     "sed -i 's/safe_to_bootstrap: 0/safe_to_bootstrap: 1/' /var/lib/mysql/grastate.dat"
312 )
313
314
315 #####
316 ##### TiDB
317 #####
318
319 cmd_install_tidb_server = "yum install tidb tidb-client tidb-ctl -y"
320 cmd_install_pd = "yum install pd -y"
321 cmd_install_tikv = "yum install tikv -y"
322
323 cmd_make_cluster_yaml=""
324 echo "
325 # For more information about the format of the tiup cluster topology file, consult
326 #
327 https://docs.pingcap.com/tidb/stable/production-deployment-using-tiup#step-3-initialize-c
luster-topology-file
328
329 # # Global variables are applied to all deployments and used as the default value of
330 # # the deployments if a specific deployment value is missing.
331 global:
332     # # The OS user who runs the tidb cluster.
333     user: \"tidb\"
334     # # SSH port of servers in the managed cluster.
335     ssh_port: 22
336     # # Storage directory for cluster deployment files, startup scripts, and configuration
337     # # files.
338     deploy_dir: \"/tidb-deploy\"
339     # # TiDB Cluster data storage directory
340     data_dir: \"/tidb-data\"
341     # # Supported values: \"amd64\", \"arm64\" (default: \"amd64\")
342     arch: \"amd64\"
343
344 pd_servers:
345     - host: 172.20.0.2
346
347 tidb_servers:
348     - host: 172.20.0.3
349
350 tikv_servers:
351     - host: 172.20.0.4
352
353 monitoring_servers:
354     - host: 172.20.0.5
355
356 grafana_servers:
357     - host: 172.20.0.6
358
359 " > cluster.yaml
360 ""
361
362 cmd_start_tidb_server = "tidb-server --config=tidb-server.toml"
363 cmd_start_pd_server = "pd-server --config=pd.toml"
364 cmd_start_tikv_server = "tikv-server --config=tikv.toml"
365
366 cmd_enable_tidb_server = "systemctl enable tidb; systemctl start tidb"
367 cmd_enable_pd_server = "tiup cluster start pd --config=pd.toml"
368 cmd_enable_tikv_server = "tiup cluster start tikv --config=tikv.toml"
369
370 # node status
371 cmd_update_tidb_node_status = (
372     "sed -i 's/status: 0/status: 1/' /var/lib/tidb/tidbstate.dat"
373 )
374
375 ## Firewall##

```

```

375 # fire_wall_add_pd_port = f"firewall-cmd --zone=public --add-port={server.pd_port}/tcp
--permanent"
376 # fire_wall_add_tikv_port = f"firewall-cmd --zone=public
--add-port={server.tikv_port}/tcp --permanent"
377 # fire_wall_add_tidb_port = f"firewall-cmd --zone=public
--add-port={server.tidb_port}/tcp --permanent"
378 # fire_wall_add_tispark_port = f"firewall-cmd --zone=public
--add-port={server.tispark_port}/tcp --permanent"
379 # fire_wall_reload = "firewall-cmd --reload"
380
381
382 # Set up TiDB cluster
383 cmd_setup_tidb_cluster = "tidb-ctl create-cluster"
384
385 # Stop TiDB
386 cmd_stop_tidb = "systemctl stop tidb"
387
388 # Configure TiDB firewall
389 cmd_configure_tidb_firewall_4000 = "firewall-cmd --permanent --add-port=4000/tcp"
390 cmd_configure_tidb_firewall_10080 = "firewall-cmd --permanent --add-port=10080/tcp"
391 cmd_reload_firewall = "firewall-cmd --reload"
392
393 cmd_install_firewall = "yum install firewalld -y"
394 #cmd_tidb_configure_firewall = "ufw allow from 0.0.0.0 to any port 4000"
395
396
397
398 #####
399 ##### Keepalived
400 #####
401
402 cmd_keepalive_install="yum -y install keepalived"
403 cmd_keepalive_stop="systemctl stop keepalived"
404 cmd_keepalive_enable="systemctl enable --now keepalived"
405 cmd_keepalive_conf_backup="mv /etc/keepalived/keepalived.conf
/etc/keepalived/keepalived.conf.bak"
406
407
408 #####
409 ##### Fail2ban
410 #####
411
412 cmd_install_Fail2ban = '''yum -y install fail2ban fail2ban-systemd
yum update -y selinux-policy*
'''
413
414 cmd_configure_fail2ban = '''cp -pf /etc/fail2ban/jail.conf /etc/fail2ban/jail.local
sed -i "/#ignoreip =/s/^#/g" /etc/fail2ban/jail.local
sed -i "285i enabled = true" /etc/fail2ban/jail.local
sed -i "286i #action = firewallcmd-ipset" /etc/fail2ban/jail.local
sed -i "287i maxretry = 5" /etc/fail2ban/jail.local
sed -i "288i bantime = 86400" /etc/fail2ban/jail.local
'''
415
416 cmd_enable_fail2ban = 'systemctl enable fail2ban && systemctl start fail2ban'
417 cmd_remove_fail2ban = '''systemctl stop fail2ban && systemctl disable fail2ban
rpm -e --nodeps `rpm -aq | grep -i fail2ban`
'''
418
419 fail2ban_restart_cmd = 'systemctl restart fail2ban'
420
421
422
423
424 #####
425 ##### ClamAV
426 #####
427
428
429
430
431 cmd_install_ClamAV = 'yum -y install clamav-server clamav-data clamav-update
clamav-filesystem clamav clamav-scanner-systemd clamav-devel clamav-lib
clamav-server-systemd'

```

```

435 cmd_uninstall_ClamAV = '''systemctl stop clamd@scan && systemctl disable clamd@scan
436 rpm -e --nodeps `rpm -aq | grep -i clamav`
437 '''
438 clamav_selinux_config="setsebool -P antivirus_can_scan_system 1 && setsebool -P
439 clamd_use_jit 1"
440 clamav_config='''sed -i -e 's/^Example/#Example/' /etc/clamd.d/scan.conf
441 sed -i "/clamd.sock/s/^#//g" /etc/clamd.d/scan.conf
442 sed -i -e "s/^Example/#Example/" /etc/freshclam.conf
443 freshclam
444 '''
445 clamav_enable_cmd="systemctl start clamd@scan && systemctl enable clamd@scan"
446 clamav_scan="freshclam && clamscan -r /"
447 clamav_restart_cmd="systemctl restart clamd@scan"
448
449 #####
450 ##### Lynis audit
451 #####
452
453 cmd_install_Lynis = 'yum -y install lynis'
454 cmd_start_Lynis = 'lynis audit system'
455 cmd_stop_Lynis = 'killall -9 lynis'
456 cmd_remove_Lynis = 'rpm -e --nodeps `rpm -aq | grep -i lynis`'
457
458
459 #####
460 ##### Hosts allow deny
461 #####
462
463 ssh_touch_files = 'touch /etc/hosts.{allow,deny}'
464 ssh_add_to_allow_list = 'echo "sshd: {}" >> /etc/hosts.allow'
465
466 #ssh_deny_hosts = 'echo "sshd: ALL" >> /etc/hosts.deny' #Deny ALL hosts
467 ssh_deny_hosts = 'echo "sshd: " >> /etc/hosts.deny'
468
469
470 #####
471 ##### Inotify
472 #####
473
474 cmd_install_Inotify = '''
475 yum makecache
476 yum -y install inotify-tools
477 '''
478 cmd_remove_Inotify = 'yum -y remove inotify-tools'
479
480
481 #####
482 ##### Rootkit hunter
483 #####
484
485 cmd_download_RKhunter = '''wget
486 http://downloads.sourceforge.net/project/rkhunter/rkhunter/1.4.6/rkhunter-1.4.6.tar.gz
487 tar zxvf rkhunter-1.4.6.tar.gz
488 '''
489 cmd_install_RKhunter = 'cd rkhunter-1.4.6 && bash installer.sh --layout default
490 --install'
491 cmd_update_RKhunter = '/usr/local/bin/rkhunter --update && /usr/local/bin/rkhunter
492 --propupd'
493 cmd_run_RKhunter = 'rkhunter --check --skip-keypress'
494 cmd_remove_RKhunter = 'rm -rf /usr/local/bin/rkhunter && rm -rf /root/rkhunter-1.4.6*'
495
496 # ModEvasive
497 cmd_install_ModEvasive = 'yum install mod_evasive -y'
498 cmd_remove_ModEvasive = '''yum -y remove mod_evasive
499 systemctl restart httpd
500 '''

```



```

498
499
500 #####
501 ##### Modsecurity
502 #####
503
504 cmd_install_ModSecurity = 'yum install mod_security -y'
505 cmd_install_rule_set = 'yum -y install mod_security_crs'
506 cmd_remove_ModSecurity = '''yum -y remove mod_security
507 systemctl restart httpd
508 '''
509 apache_reload_cmd = 'systemctl reload httpd'
510
511
512 #####
513 ##### Haproxy
514 #####
515
516 cmd_install_haproxy="yum -y install haproxy"
517 cmd_chmod="cd /etc/firewalld/services && restorecon haproxy.xml && chmod 640 haproxy.xml"
518 cmd_haproxy_add_service="firewall-cmd --permanent --add-service=haproxy"
519 cmd_firewalld_reload="firewall-cmd --reload"
520 cmd_setsebool_1="setsebool -P haproxy_connect_any 1"
521 cmd_systemctl_status="systemctl status haproxy.service -l --no-pager"
522 cmd_haproxy_stop="systemctl stop haproxy"
523 cmd_haproxy_enable="systemctl enable --now haproxy"
524
525
526 #####
527 ##### SYSTAT
528 #####
529
530 cmd_SYSSTAT_install_path = 'rpm -q --filesbypkg sysstat'
531 cmd_SYSSTAT_file_list = 'find / -name "'sysstat'" '
532
533 cmd_install_SYSSTAT = 'yum -y install sysstat'
534 cmd_enable_SYSSTAT = ''
535 cmd_start_SYSSTAT = ''
536
537
538 #####
539 ##### ModEvasive
540 #####
541
542 cmd_install_ModEvasive = 'yum install mod_evasive -y'
543 cmd_remove_ModEvasive = '''yum -y remove mod_evasive
544 systemctl restart httpd
545 '''
546
547
548 #####
549 ##### Mailu
550 #####
551
552 cmd_show_Mailu_file_list = 'find / -type d -name mailu'
553
554 cmd_show_Docker_Mailu_files = 'docker container ls' #loop to see if if 'mailu' in the
text
555 cmd_install_Docker_Mailu = ''
556
557 cmd_start_docker_compose_Mailu = 'docker-compose -p mailu up -d'
558
559 cmd_mailu_create_admin_user = ' docker-compose -p mailu exec -T admin flask mailu admin
$MAIL_SERVER_AMDIN_USR $MAIL_SERVER_DOMAIN "$MAIL_SERVER_AMDIN_PASS" '
560
561

```